

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Original) A media platform, comprising:
  - a media provider module to receive application requests for DMA slot allocation;
  - a media group module to receive direct memory access (DMA) slot allocation instructions from the media provider module;
  - a first driver associated with a telecom media card (TMC) to receive instructions from the media provider module to provide media data traffic connections on the TMC; and
  - a TMC proxy module coupled to a second driver associated with the TMC, the TMC proxy module to receive DMA slot allocation instructions from the media group module and to communicate DMA slot and buffer address instructions to the second driver to connect DMA slots.
2. (Original) The media platform of claim 1, wherein the media provider module and the media group module are provided in a user space.
3. (Original) The media platform of claim 1, wherein the DMA slots provide media channels and bus slots access to a direct memory access (DMA) module coupled between the TMC proxy module and the TMC.
4. (Original) The media platform of claim 1, wherein the media group module communicates instructions to the media provider module to release a DMA slot.

5. (Original) The media platform of claim 1, wherein the media provider module includes a first application program interface (API) library, the media provider module to retrieve APIs from the first API library to execute requests for connections to set up and tear down media channel, bus slot, and direct memory access (DMA) slot connections.
6. (Original) The media platform of claim 5, wherein the first API library is a Java Native Interface (JNI) library.
7. (Original) The media platform of claim 1, wherein the media group module includes a second API library, the media group module to retrieve APIs from the second API library to allocate DMA slots.
8. (Original) The media platform of claim 7, wherein the second API library is a JNI library and includes C++ programming instructions.
9. (Original) The media platform of claim 1, wherein the TMC proxy module is further coupled to a digital signal processor (DSP) software module to process media data traffic with the TMC.
10. (Original) The media platform of claim 1, wherein the TMC includes T1 media channels and H100 bus slots for integrated services digital network connections.
11. (Original) The media platform of claim 1, wherein the TMC includes E1 media channels.
12. (Original) The media platform of claim 1, wherein the TMC includes J1 media channels.

13. (Original) The media platform of claim 1, wherein the media platform includes a Linux based operating system.

14. (Original) The media platform of claim 1, wherein the media platform includes a Unix based operating system.

15. (Original) A media platform, comprising:

a media provider module including a first application program interface (API) library, the media provider module to retrieve APIs from the first API library to allocate application requests with a first driver associated with a telecom media card (TMC);

a media group module including a second API library, the media group module to retrieve APIs from the second API library based on DMA slot allocation instructions received from the media provider module; and

a TMC proxy module coupled to a second driver associated with the TMC, the TMC proxy module to receive DMA slot allocation instructions from the media group module.

16. (Original) The media platform of claim 15, further including a digital signal processor (DSP) software module coupled to the TMC proxy module to process media data traffic, and wherein the TMC proxy module provides media data traffic access between the DSP software module and a DMA module coupled between the TMC proxy module and the TMC.

17. (Original) The media platform of claim 16, wherein the DSP software module and a portion of the TMC proxy module are located in a user space.

18. (Original) The media platform of claim 16, wherein media channels and bus slots are connected to DMA slots of the DMA module based on DMA slot and

buffer address instructions communicated to the second driver by the TMC proxy module.

19. (Original) The media platform of claim 15, wherein the APIs from the first API library are used to execute the requests for connections to set up and tear down media channel, bus slot, and DMA slot connections.

20. (Original) The media platform of claim 15, wherein the APIs from the second API library are used to allocate DMA slots, and wherein the media group module further includes program instructions to release DMA slots to the media provider module.

21. (Original) A media platform, comprising:  
a memory including a telecommunication program application;  
a processor couple to the memory to execute the program application;  
a telecom media card (TMC) including media channels and bus slots, the TMC to receive media channel, bus slot, and direct memory access (DMA) slot connection requests associated with the program application;  
a direct memory access (DMA) module coupled to the TMC;  
a digital signal processor (DSP) software module coupled to the DMA module;  
means for allocating DMA slots to media channels and bus slots on the TMC; and  
means for providing media data traffic access between the DSP software module and a DMA module.

22. (Original) The media platform of claim 21, wherein means for allocating DMA slots to media channels and bus slots includes providing a first set of APIs to execute instructions to set up and tear down connections with T1 media channel and

H100 bus slot connections through a first driver associated with the TMC and to execute DMA slot allocation instructions.

23. (Original) The media platform of claim 21, wherein means for providing media data traffic access between the DSP software module and a DMA module includes receiving DMA slot allocation instructions on a module having a second set of APIs, the second set of APIs to communicate DMA slot allocation instructions to the TMC and to release DMA slots when no longer being used.

24. (Original) The media platform of claim 23, further including a TMC proxy module coupled to a second driver associated with the TMC, the TMC proxy module to receive DMA slot allocation instructions from the module having a second set of APIs and to communicate DMA slot and buffer address instructions to the second driver.

25. (Original) A method for allocating direct memory access (DMA) slots, comprising:

- receiving DMA requests for connecting media data traffic to DMA slots of a DMA memory module;

- determining available DMA slots from a pool of available DMA slots and assigning DMA slots;

- releasing DMA slots to the pool of available DMA slots when a DMA slot is no longer being used; and

- using a TMC proxy to connect media data traffic between the DMA memory module and a DSP software module based on assigned DMA slots.

26. (Original) The method of claim 25, further including:

- providing a first set of APIs to a first module, the first set of APIs to receive requests for DMA slot allocation and to execute switching connections for media data traffic on a telecom media card (TMC);

providing a second set of APIs to a second module, the second set of APIs to receive DMA slot allocation instructions from the first module and to execute DMA slot allocation instructions through the proxy, and the second set of APIs to release DMA slots to the first module based on feedback from the proxy; and

communicating DMA slot allocation instructions from the proxy to the TMC to establish connections between DMA slots of the DMA memory module and the media channels and bus slots on the TMC, wherein the proxy includes instructions to connect media data traffic between the DSP software module and the DMA memory module.

27. (Original) The method of claim 26, further including providing a Java Native Interface (JNI) library as the first set of APIs.

28. (Original) The method of claim 26, further including providing a JNI library and C++ programming instructions as the second set of APIs.

29. (Original) The method of claim 25, further including connecting E1 media channels and H100 bus slots for integrated services digital network connections to the DMA slots.

30. (Original) The method of claim 25, further including connecting T1 media channels and H100 bus slots for integrated services digital network connections to the DMA slots.

31. (Original) The method of claim 25, further including connecting J1 media channels and H100 bus slots for integrated services digital network connections to the DMA slots.

32. (Original) A method for allocating direct memory access (DMA) slots, comprising:

tracking a pool of available DMA slots in a user space media provider module;

using a first set of APIs associated with the media provider module to execute instructions to connect media data traffic to available DMA slots; and

communicating assigned DMA slots to a media group module in the user space, the media group module having a second set of APIs to execute instructions to connect assigned DMA slots to a digital signal processing module.

33. (Original) The method of claim 32, further including assigning available DMA slots to T1 media channels and H100 bus slots.

34. (Original) A computer readable medium having a program to cause a device to perform a method, comprising:

receiving DMA requests for connecting media data traffic to DMA slots of a DMA memory module;

determining available DMA slots from a pool of available DMA slots and assigning DMA slots;

releasing DMA slots to the pool of available DMA slots when a DMA slot is no longer being used; and

using a TMC proxy to connect media data traffic between the DMA memory module and a DSP software module based on assigned DMA slots.